Chapter 13

ARTIFICIAL IMMUNE SYSTEMS

U. Aickelin
University of Nottingham, UK

D. Dasgupta
University of Memphis, Memphis, TN 38152, USA

13.1 INTRODUCTION

The biological immune system is a robust, complex, adaptive system that defends the body from foreign pathogens. It is able to categorize all cells (or molecules) within the body as self-cells or nonself cells. It does this with the help of a distributed task force that has the intelligence to take action from a local and also a global perspective using its network of chemical messengers for communication. There are two major branches of the immune system. The innate immune system is an unchanging mechanism that detects and destroys certain invading organisms, whilst the adaptive immune system responds to previously unknown foreign cells and builds a response to them that can remain in the body over a long period of time. This remarkable information processing biological system has caught the attention of computer science in recent years.

A novel computational intelligence technique, inspired by immunology, has emerged, known as Artificial Immune Systems. Several concepts from immunology have been extracted and applied for the solution of real-world science and engineering problems. In this tutorial, we briefly describe the immune system metaphors that are relevant to existing Artificial Immune System methods. We then introduce illustrative real-world problems and give a step-by-step algorithm walkthrough for one such problem. A comparison of Artificial Immune Systems to other well-known algorithms, areas for future work, tips and tricks and a list of resources round the tutorial off. It should be noted that as Artificial Immune Systems is still a young and evolving field, there is not yet a

fixed algorithm template and hence actual implementations may differ somewhat from time to time and from those examples given here.

13.2 OVERVIEW OF THE BIOLOGICAL IMMUNE SYSTEM

The biological immune system is an elaborate defense system which has evolved over millions of years. While many details of the immune mechanisms (innate and adaptive) and processes (humoral and cellular) are yet unknown (even to immunologists), it is, however, well known that the immune system uses multilevel (and overlapping) defense both in parallel and sequential fashion. Depending on the type of the pathogen, and the way it gets into the body, the immune system uses different response mechanisms (differential pathways) either to neutralize the pathogenic effect or to destroy the infected cells. A detailed overview of the immune system can be found in many textbooks, such as Kubi (2002). The immune features that are particularly relevant to our tutorial are matching, diversity and distributed control. Matching refers to the binding between antibodies and antigens. Diversity refers to the fact that, in order to achieve optimal antigen space coverage, antibody diversity must be encouraged (see Hightower et al., 1995). Distributed control means that there is no central controller; rather, the immune system is governed by local interactions between immune cells and antigens.

Two of the most important cells in this process are white blood cells, called T-cells and B-cells. Both of these originate in the bone marrow, but T-cells pass on to the thymus to mature, before circulating in the blood and lymphatic vessels.

The T-cells are of three types: helper T-cells which are essential to the activation of B-cells, killer T-cells which bind to foreign invaders and inject poisonous chemicals into them causing their destruction, and suppressor T-cells which inhibit the action of other immune cells thus preventing allergic reactions and autoimmune diseases.

B-cells are responsible for the production and secretion of antibodies, which are specific proteins that bind to the antigen. Each B-cell can only produce one particular antibody. The antigen is found on the surface of the invading organism and the binding of an antibody to the antigen is a signal to destroy the invading cell as shown in Fig. 13.1.

As mentioned above, the human body is protected against foreign invaders by a multi-layered system. The immune system is composed of physical barriers such as the skin and respiratory system; physiological barriers such as destructive enzymes and stomach acids; and the immune system, which can be broadly viewed as of two types: in-

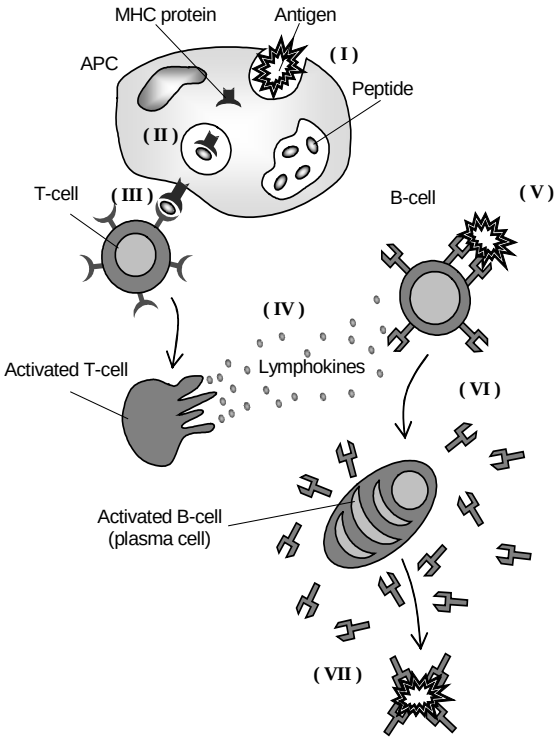

Figure 13.1. Pictorial representation of the essence of the acquired immune system mechanism (taken from de Castro and van Zuben (1999): the invade enters the body and activates T-cells, which then in IV activate the B-cells; V is the antigen matching, VI the antibody production and VII the antigen's destruction.

nate (non-specific) immunity and adaptive (specific) immunity, which are inter-linked and influence each other. Adaptive immunity can again be subdivided into two types: humoral immunity and cell-mediated immunity.

Innate immunity is present at birth. Physiological conditions such as pH, temperature and chemical mediators provide inappropriate living conditions for foreign organisms. Also, micro-organisms are coated with antibodies and/or complementary products (opsonization) so that they are easily recognized. Extracellular material is then ingested by macrophages by a process called phagocytosis. Also, T_{DH}-cells influence the phagocytosis of macrophages by secreting certain chemical messengers called lymphokines. The low levels of sialic acid on foreign antigenic surfaces make C_3b bind to these surfaces for a long time and thus acti-

vate alternative pathways. Thus MAC is formed, which punctures the cell surfaces and kills the foreign invader.

Adaptive immunity is the main focus of interest here as learning, adaptability, and memory are important characteristics of adaptive immunity. It is subdivided under two heads: humoral immunity and cell-mediated immunity:

1 Humoral immunity is mediated by antibodies contained in body fluids (known as humors). The humoral branch of the immune system involves interaction of B-cells with antigen and their subsequent proliferation and differentiation into antibody-secreting plasma cells. Antibody functions as the effectors of the humoral response by binding to antigen and facilitating its elimination. When an antigen is coated with antibody, it can be eliminated in several ways. For example, antibody can cross-link the antigen, forming clusters that are more readily ingested by phagocytic cells. Binding of antibody to antigen on a micro-organism also can activate the complement system, resulting in lysis of the foreign organism.

2 Cellular immunity is cell-mediated; effector T-cells generated in response to antigen are responsible for cell-mediated immunity. Cytotoxic T-lymphocytes (CTLs) participate in cell-mediated immune reactions by killing altered self-cells; they play an important role in the killing of virus-infected and tumor cells. Cytokines secreted by T_{DH} can mediate the cellular immunity, and activate various phagocytic cells, enabling them to phagocytose and kill micro-organisms more effectively. This type of cell-mediated immune response is especially important in host defense against intracellular bacteria and protozoa.

Whilst there is more than one mechanism at work (for more details see Farmer et al., 1986; Kubi, 2002; Jerne, 1973), the essential process is the matching of antigen and antibody, which leads to increased concentrations (proliferation) of more closely matched antibodies. In particular, idiotypic network theory, negative selection mechanism, and the "clonal selection" and "somatic hypermutation" theories are primarily used in Artificial Immune System models.

13.2.1 Immune Network Theory

The immune network theory was proposed by Jerne (1973). The hypothesis was that the immune system maintains an idiotypic network of interconnected B-cells for antigen recognition. These cells both stimulate and suppress each other in certain ways that lead to the stabilization

of the network. Two B-cells are connected if the affinities they share exceed a certain threshold, and the strength of the connection is directly proportional to the affinity they share.

13.2.2 Negative Selection Mechanism

The purpose of negative selection is to provide tolerance for self-cells. It deals with the immune system's ability to detect unknown antigens while not reacting to the self-cells. During the generation of T-cells, receptors are made through a pseudo-random genetic rearrangement process. Then, they undergo a censoring process in the thymus, called the negative selection. There, T-cells that react against self-proteins are destroyed; thus, only those that do not bind to self-proteins are allowed to leave the thymus. These matured T-cells then circulate throughout the body to perform immunological functions and protect the body against foreign antigens.

13.2.3 Clonal Selection Principle

The clonal selection principle describes the basic features of an immune response to an antigenic stimulus. It establishes the idea that only those cells that recognize the antigen proliferate, thus being selected against those that do not. The main features of the clonal selection theory are that

1 the new cells are copies of their parents (clone) subjected to a mutation mechanism with high rates (somatic hypermutation);

2 elimination of newly differentiated lymphocytes carrying self-reactive receptors;

3 proliferation and differentiation on contact of mature cells with antigens.

When an antibody strongly matches an antigen the corresponding B-cell is stimulated to produce clones of itself that then produce more antibodies. This (hyper) mutation, is quite rapid, often as much as "one mutation per cell division" (de Castro and Von Zuben, 1999). This allows a very quick response to the antigens. It should be noted here that in the Artificial Immune System literature, often no distinction is made between B-cells and the antibodies they produce. Both are subsumed under the word 'antibody' and statements such as mutation of antibodies (rather than mutation of B-cells) are common.

There are many more features of the immune system, including adaptation, immunological memory and protection against auto-immune at-

tacks, not discussed here. In the following sections, we will revisit some important aspects of these concepts and show how they can be modeled in "artificial" immune systems and then used to solve real-world problems. First, let us give an overview of typical problems that we believe are amenable to being solved by artificial immune systems.

13.3 ILLUSTRATIVE PROBLEMS

13.3.1 Intrusion Detection Systems

Anyone keeping up-to-date with current affairs in computing can confirm numerous cases of attacks made on computer servers of well-known companies. These attacks range from denial-of-service attacks to extracting credit-card details and sometimes we find ourselves thinking "haven't they installed a firewall"? The fact is they often have a firewall. A firewall is useful, indeed often essential, but current firewall technology is insufficient to detect and block all kinds of attacks.

On ports that need to be open to the internet, a firewall can do little to prevent attacks. Moreover, even if a port is blocked from internet access, this does not stop an attack from inside the organization. This is where intrusion detection systems come in. As the name suggests, intrusion detection systems are installed to identify (potential) attacks and to react by usually generating an alert or blocking the unscrupulous data.

The main goal of intrusion detection systems is to detect unauthorized use, misuse and abuse of computer systems by both system insiders and external intruders. Most current intrusion detection systems define suspicious signatures based on known intrusions and probes. The obvious limit of this type of intrusion detection systems is its failure in detecting previously unknown intrusions. In contrast, the human immune system adaptively generates new immune cells so that it is able to detect previously unknown and rapidly evolving harmful antigens (Forrest et al., 1994). Thus the challenge is to emulate the success of the natural systems.

13.3.2 Data Mining—Collaborative Filtering and Clustering

Collaborative filtering is the term for a broad range of algorithms that use similarity measures to obtain recommendations. The best-known example is probably the "people who bought this also bought" feature of the internet company Amazon (2004). However, any problem domain where users are required to rate items is amenable to collaborative filter-

ing techniques. Commercial applications are usually called recommender systems (Resnick and Varian, 1997). A canonical example is movie recommendation.

In traditional collaborative filtering, the items to be recommended are treated as "black boxes". That is, your recommendations are based purely on the votes of other users, and not on the content of the item. The preferences of a user, usually a set of votes on an item, comprise a user profile, and these profiles are compared in order to build a neighborhood. The key decision is what similarity measure is used. The most common method to compare two users is a correlation-based measure like Pearson or Spearman, which gives two neighbors a matching score between -1 and 1. The canonical example is the k-nearest-neighbor algorithm, which uses a matching method to select k reviewers with high similarity measures. The votes from these reviewers, suitably weighted, are used to make predictions and recommendations.

The evaluation of a collaborative filtering algorithm usually centers on its accuracy. There is a difference between prediction (given a movie, predict a given user's rating of that movie) and recommendation (given a user, suggest movies that are likely to attract a high rating). Prediction is easier to assess quantitatively but recommendation is a more natural fit to the movie domain. A related problem to collaborative filtering is that of clustering data or users in a database. This is particularly useful in very large databases, which have become too large to handle. Clustering works by dividing the entries of the database into groups, which contain people with similar preferences or in general data of similar type.

13.4 ARTIFICIAL IMMUNE SYSTEMS BASIC CONCEPTS

13.4.1 Initialization/Encoding

To implement a basic artificial immune system, four decisions have to be made: encoding, similarity measure, selection and mutation. Once an encoding has been fixed and a suitable similarity measure is chosen, the algorithm will then perform selection and mutation, both based on the similarity measure, until stopping criteria are met. In this section, we will describe each of these components in turn.

Along with other heuristics, choosing a suitable encoding is very important for the algorithm's success. Similar to genetic algorithms, there is close inter-play between the encoding and the fitness function (the latter is in artificial immune systems referred to as the "matching" or

"affinity" function). Hence both ought to be thought about at the same time. For the current discussion, let us start with the encoding.

First, let us define what we mean by antigen and antibody in the context of an application domain. Typically, an antigen is the target or solution, e.g. the data item we need to check to see if it is an intrusion, or the user that we need to cluster or make a recommendation for. The antibodies are the remainder of the data, e.g. other users in the data base, a set of network traffic that has already been identified, etc. Sometimes there can be more than one antigen at a time and there are usually a large number of antibodies present simultaneously.

Antigens and antibodies are represented or encoded in the same way. For most problems the most obvious representation is a string of numbers or features, where the length is the number of variables, the position is the variable identifier and the value is the value (could be binary or real) of the variable. For instance, in a five-variable binary problem, an encoding could look like this: (10010).

We have previously mentioned data mining and intrusion detection applications. What would an encoding look like in these cases? For data mining, let us consider the problem of recommending movies. Here the encoding has to represent a user's profile with regards to the movies he has seen and how much he has (dis)liked them. A possible encoding for this could be a list of numbers, where each number represents the "vote" for an item. Votes could be binary, e.g. Did you visit this web page? (Morrison and Aickelin, 2002), but can also be integers in a range (say $[0, 5]$: i.e. 0, did not like the movie at all; 5, liked it very much).

Hence for the movie recommendation, a possible encoding is

$$User = \{\{id_1, score_1\}, \{id_2, score_2\} \ldots \{id_n, score_n\}\}$$

Where *id* corresponds to the unique identifier of the movie being rated and score to this user's score for that movie. This captures the essential features of the data available (Cayzer and Aickelin, 2002a).

For intrusion detection, the encoding may be to encapsulate the essence of each data packet transferred, e.g. [<protocol><source ip><source port><destination ip><destination port>]

example: [<tcp> <113.112.255.254><any><108.200.111.12><25>]

which represents an incoming data packet sent to port 25. In these scenarios, wildcards like "any port" are also often used.

13.4.2 Similarity or Affinity Measure

As mentioned above, the similarity measure or matching rule is one of the most important design choices in developing an artificial immune system algorithm, and is closely coupled to the encoding scheme.

Two of the simplest matching algorithms are best explained using binary encoding. Consider the strings (00000) and (00011). If one does a bit-by-bit comparison, the first three bits are identical and hence we could give this pair a matching score of 3. In other words, we compute the opposite of the Hamming distance (which is defined as the number of bits that have to be changed in order to make the two strings identical).

Now consider the pair (00000) and (01010). Again, simple bit matching gives us a similarity score of 3. However, the matching is quite different as the three matching bits are not connected. Depending on the problem and encoding, this might be better or worse. Thus, another simple matching algorithm is to count the number of continuous bits that match and return the length of the longest matching as the similarity measure. For the first example above this would still be 3; for the second example it would be 1.

If the encoding is non-binary, e.g. real variables, there are even more possibilities to compute the "distance" between the two strings, for instance we could compute the geometrical (Euclidian) distance.

For data mining problems, similarity often means "correlation". Take the movie recommendation problem as an example and assume that we are trying to find users in a database that are similar to the key user who's profile were are trying to match in order to make recommendations. In this case, what we are trying to measure is how similar are the two users' tastes. One of the easiest ways of doing this is to compute the Pearson correlation coefficient between the two users, i.e. if the Pearson measure is used to compare two user's u and v:

$$r = \frac{\sum_{i=1}^{n}(u_i - \bar{u})(v_i - \bar{v})}{\sqrt{\sum_{i=1}^{n}(u_i - \bar{u})^2 \sum_{i=1}^{n}(v_i - \bar{v})^2}} \tag{4.1}$$

where u and v are users, n is the number of overlapping votes (i.e. movies for which both u and v have voted), u_i is the vote of user u for movie i and $\bar{u}$ is the average vote of user u over all films (not just the overlapping votes). The measure is amended to default to a value of 0 if the two users have no films in common. During our research reported in Cayzer and Aickelin (2002a, 2002b) we also found it useful to introduce a penalty

parameter (as in penalties in genetic algorithms) for users who only have very few films in common, which in essence reduces their correlation.

The outcome of this measure is a value between -1 and 1, where values close to 1 mean strong agreement, values near to -1 mean strong disagreement and values around 0 mean no correlation. From a data mining point of view, those users who score either 1 or -1 are the most useful and hence will be selected for further treatment by the algorithm.

For other applications, "matching" might not actually be beneficial and hence those items that match might be eliminated. This approach is known as "negative selection" and mirrors what is believed to happen during the maturation of B-cells who have to learn not to "match" our own tissues as otherwise we would be subject to auto-immune diseases.

Under what circumstance would a negative selection algorithm be suitable for an artificial immune system implementation? Consider the case of intrusion detection as solved by Hofmeyr and Forrest (2000). One way of solving this problem is by defining a set of "self", i.e. a trusted network, our company's computers, known partners, etc. During the initialization of the algorithm, we would then randomly create a large number of "detectors", i.e. strings that look similar to the sample intrusion detection system encoding given above. We would then subject these detectors to a matching algorithm that compares them to our "self". Any matching detector would be eliminated and hence we select those that do not match (negative selection). All non-matching detectors will then form our final detector set. This detector set is then used in the second phase of the algorithm to continuously monitor all network traffic. Should a match be found now the algorithm would report this as a possible alert or "nonself". There are a number of problems with this approach, which we discuss further in Section 7.

13.4.3 Negative, Clonal or Neighborhood Selection

The meaning of this step differs depending on the exact problem the Artificial Immune Systems is applied to. We have already described the concept of negative selection. For the film recommender, choosing a suitable neighborhood means choosing good correlation scores and hence we will perform "positive" selection. How would the algorithm use this?

Consider the artificial immune system to be empty at the beginning. The target user is encoded as the antigen, and all other users in the database are possible antibodies. We add the antigen to the artificial immune system and then we add one candidate antibody at a time. Antibodies will start with a certain concentration value. This value

decreases over time (death rate), similar to the evaporation in ant systems. Antibodies with a sufficiently low concentration are removed from the system, whereas antibodies with a high concentration may saturate. However, an antibody can increase its concentration by matching the antigen: the better the match the higher the increase (a process called stimulation). The process of stimulation or increasing concentration can also be regarded as "cloning" if one thinks in a discrete setting. Once enough antibodies have been added to the system, it starts to iterate a loop of reducing concentration and stimulation until at least one antibody drops out. A new antibody is added and the process is repeated until the artificial immune system is stabilized, i.e. there are no more drop-outs for a certain period of time.

Mathematically, at each step (iteration) an antibody's concentration is increased by an amount dependent on its matching to each antigen. In the absence of matching, an antibody's concentration will slowly decrease over time. Hence an artificial immune system iteration is governed by the following equation, based on Farmer et al. (1986):

$$\begin{aligned} \frac{\mathrm{d}x_i}{\mathrm{d}t} &= \left[\begin{pmatrix} antigens \\ recognized \end{pmatrix} - \begin{pmatrix} death \\ rate \end{pmatrix} \right] \\ &= \left[k_2 \left(\sum_{j=1}^{N} m_{ji} x_i y_j \right) - k_3 x_i \right] \end{aligned}$$

where N is the number of antigens, x_i is the concentration of antibody i, y_j is the concentration of antigen j, k_2 is the stimulation effect and k_3 is the death rate, and m_{ji} is the matching function between antibody i and antibody (or antigen) j.

The following pseudo-code summarizes the artificial immune system of the movie recommender:

```
Initialize Artificial Immune Systems
Encode user for whom to make predictions as antigen Ag
WHILE (Artificial Immune Systems not Full) & (More Antibodies) DO
   Add next user as an antibody Ab
   Calculate matching scores between Ab and Ag
   WHILE (Artificial Immune Systems at full size) & (Artificial Immune
   Systems not Stabilized) DO
      Reduce Concentration of all Abs by a fixed amount
      Match each Ab against Ag and stimulate as necessary
   OD
OD
Use final set of Antibodies to produce recommendation.
```

For example, the artificial immune system is considered stable after iterating for ten iterations without changing in size. Stabilization thus means that a sufficient number of "good" neighbors have been identified and therefore a prediction can be made. "Poor" neighbors would be expected to drop out of the artificial immune system after a few iterations. Once the artificial immune system has stabilized using the above algorithm, we use the antibody concentration to weigh the neighbors and then perform a weighted average type recommendation.

13.4.4 Somatic Hypermutation

The mutation most commonly used in artificial immune systems is very similar to that found in genetic algorithms, e.g. for binary strings bits are flipped, for real value strings one value is changed at random, or for others the order of elements is swapped. In addition, the mechanism is often enhanced by the somatic idea, i.e. the closer the match (or the less close the match, depending on what we are trying to achieve), the more (or less) disruptive the mutation.

However, mutating the data might not make sense for all problems considered. For instance, it would not be suitable for the movie recommender. Certainly, mutation could be used to make users more similar to the target; however, the validity of recommendations based on these artificial users is questionable and if over-done, we would end up with the target user itself. Hence for some problems, somatic hypermutation is not used, since it is not immediately obvious how to mutate the data sensibly such that these artificial entities still represent plausible data.

Nevertheless, for other problem domains, mutation might be very useful. For instance, taking the negative selection approach to intrusion detection, rather than throwing away matching detectors in the first phase of the algorithm, these could be mutated to save time and effort. Also, depending on the degree of matching, the mutation could be more or less strong. This was in fact one extension implemented by Hofmeyr and Forrest (2000).

For data mining problems, mutation might also be useful, if for instance the aim is to cluster users. Then the center of each cluster (the antibodies) could be an artificial pseudo-user that can be mutated at will until the desired degree of matching between the center and antigens in its cluster is reached. This is an approach implemented by de Castro and von Zuben (2002).

13.5 COMPARISON WITH GENETIC ALGORITHMS AND NEURAL NETWORKS

So far in this tutorial, both genetic algorithms and neural networks have been mentioned a number of times. In fact, they both have a number of ideas in common with artificial immune systems and Table 13.1 highlights their similarities and differences (Dasgupta, 1999). Evolutionary computation shares many elements, concepts like population, genotype phenotype mapping, and proliferation of the most fitted are present in different artificial immune system methods.

Artificial immune system models based on immune networks resemble the structures and interactions of connectionist models. Some works have pointed to the similarities and the differences between artificial immune systems and artificial neural networks (Dasgupta, 1999; de Castro and Von Zuben, 2002); de Castro has also used artificial immune systems to initialize the centers of radial basis function neural networks and to produce a good initial set of weights for feed-forward neural networks.

Some of the items in Table 13.1 are gross simplifications, both to benefit the design of the table and so as not to overwhelm the reader, and some of the points are debatable; however, we believe that the comparison is nevertheless valuable, to show exactly where artificial immune systems fit into the wider picture. The comparisons are based on a genetic algorithm (GA) used for optimization and a neural network (NN) used for classification.

13.6 EXTENSIONS OF ARTIFICIAL IMMUNE SYSTEMS

13.6.1 Idiotypic Networks—Network Interactions (Suppression)

The idiotypic effect builds on the premise that antibodies can match other antibodies as well as antigens. It was first proposed by Jerne (1973) and formalized into a model by Farmer et al. (1986). The theory is currently debated by immunologists, with no clear consensus yet on its effects in the humoral immune system (Kuby, 2002). The idiotypic network hypothesis builds on the recognition that antibodies can match other antibodies as well as antigens. Hence, an antibody may be matched by other antibodies, which in turn may be matched by yet other antibodies. This activation can continue to spread through the population and potentially has much explanatory power. It could, for example, help explain how the memory of past infections is maintained.

Table 13.1. Comparison of artificial immune systems with genetic algorithms and neural networks

	GA (Optimization)	NN (Classification)	Artificial immune systems
Components	Chromosome strings	Artificial neurons	Attribute strings
Location of components	Dynamic	Pre-defined	Dynamic
Structure	Discrete components	Networked components	Discrete/networked components
Knowledge storage	Chromosome strings	Connection strengths	Component concentration/network connections
Dynamics	Evolution	Learning	Evolution/learning
Meta-dynamics	Recruitment/elimination of components	Construction/pruning of connections	Recruitment/elimination of components
Interaction between components	Crossover	Network connections	Recognition/network connections
Interaction with environment	Fitness function	External stimuli	Recognition/objective function
Threshold activity	Crowding/sharing	Neuron activation	Component affinity

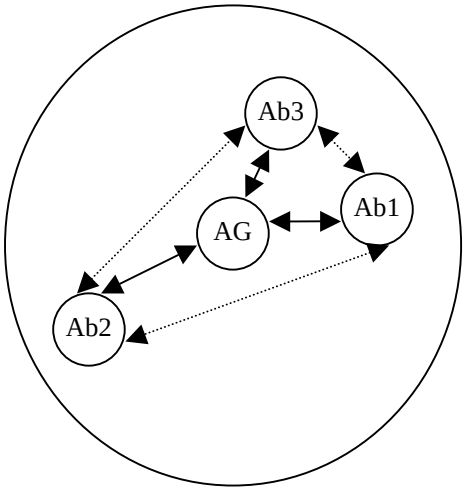

Figure 13.2. Illustration of the idiotypic effect.

Furthermore, it could result in the suppression of similar antibodies, thus encouraging diversity in the antibody pool. The idiotypic network has been formalized by a number of theoretical immunologists (Perelson and Weisbuch, 1997):

$$\begin{aligned}\frac{\mathrm{d}x_i}{\mathrm{d}t} &= c\left[\left(\begin{array}{l} antibodies \\ recognized \end{array}\right) - \left(\begin{array}{l} I\,am \\ recognized \end{array}\right) + \left(\begin{array}{l} antigens \\ recognized \end{array}\right)\right] \\ &\quad - \left(\begin{array}{l} death \\ rate \end{array}\right) \\ &= c\left[\sum_{j=1}^{N} m_{ji}x_ix_j - k_1\sum_{j-1}^{N} m_{ij}x_ix_j + \sum_{j=1}^{n} m_{ji}x_iy_j\right] - k_2x_i\end{aligned}$$

where N is the number of antibodies and n is the number of antigens, x_i (or x_j) is the concentration of antibody i (or j), y_j is the concentration of antigen j, c is a rate constant, k_1 is a suppressive effect and k_2 is the death rate, and m_{ji} is the matching function between antibody i and antibody (or antigen) j.

As can be seen from the above equation, the nature of an idiotypic interaction can be either positive or negative. Moreover, if the matching function is symmetric, then the balance between "I am recognized" and "antibodies recognized" (parameters c and k_1 in the equation) wholly determines whether the idiotypic effect is positive or negative, and we can simplify the equation. We can further simplify Eq. (1) if we only allow one antigen in the artificial immune system. In Eq. (2), the first

term is simplified as we only have one antigen, and the suppression term is normalized to allow a "like for like" comparison between the different rate constants:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = k_1 m_i x_i y - \frac{k_2}{n} \sum_{j=1}^{n} m_{ij} x_i x_j - k_3 x_i \tag{6.1}$$

where k_1 is stimulation, k_2 suppression, k_3 death rate, m_i is the correlation between antibody i and the (sole) antigen, x_i (or x_j) is the concentration of antibody i (or j), y is the concentration of the (sole) antigen, m_{ij} is the correlation between antibodies i and j, and n is the number of antibodies.

Why would we want to use the idotypic effect? Because it might provide us with a way of achieving "diversity", similar to "crowding" or "fitness sharing" in a genetic algorithm. For instance, in the movie recommender, we want to ensure that the final neighborhood population is diverse, so that we get more interesting recommendations. Hence, to use the idiotypic effect in the movie recommender system mentioned previously, the pseudo-code would be amended by adding the italicized lines as follows:

```
Initialize Artificial Immune Systems
Encode user for whom to make predictions as antigen Ag
WHILE (Artificial Immune Systems not Full) & (More Antibodies) DO
   Add next user as an antibody Ab
   Calculate matching scores between Ab and Ag and Ab and other Abs
   WHILE (Artificial Immune Systems at full size) & (Artificial Immune
   Systems not Stabilized) DO
      Reduce Concentration of all Abs by a fixed amount
      Match each Ab against Ag and stimulate as necessary
      Match each Ab against each other Ab and execute idiotypic effect
   OD
OD
Use final set of Antibodies to produce recommendation.
```

Figure 11-2 shows the idiotypic effect using dotted arrows and the standard stimulation using solid arrows. In the diagram antibodies Ab1 and Ab3 are very similar and they would have their concentrations reduced in the "iterate artificial immune systems" stage of the algorithm above.

At each iteration of the film recommendation artificial immune system the concentration of the antibodies is changed according to the formula outlined below. This will increase the concentration of antibodies that

are similar to the antigen and can allow either the stimulation, suppression, or both, of antibody–antibody interactions to have an effect on the antibody concentration. More detailed discussion of these effects on recommendation problems are contained within Cayzer and Aickelin (2002a, b).

13.6.2 Danger Theory

Over the last decade, a new theory, called the Danger Theory, has become popular amongst immunologists. Its chief advocate is Matzinger (1994, 2001, 2003). A number of advantages are claimed for this theory; not least that it provides a method of "grounding" the immune response. The theory is not complete, and there are some doubts about how much it actually changes behaviour and/or structure. Nevertheless, the theory contains enough potentially interesting ideas to make it worth assessing its relevance to artificial immune systems.

To function properly, it is not simply a question of matching in the humoral immune system. It is fundamental that only the "correct" cells are matched as otherwise this could lead to a self-destructive autoimmune reaction. Classical immunology (Kuby, 2002) stipulates that an immune response is triggered when the body encounters something nonself or foreign. It is not yet fully understood how this self–nonself discrimination is achieved, but many immunologists believe that the difference between them is learnt early in life. In particular, it is thought that the maturation process plays an important role to achieve self-tolerance by eliminating those T- and B-cells that react to self. In addition, a "confirmation" signal is required: that is, for either B-cell or T- (killer) cell activation, a T- (helper) lymphocyte must also be activated. This dual activation is further protection against the chance of accidentally reacting to self.

Danger Theory debates this point of view (for a good introduction, see Matzinger, 2003). Technical overviews can be found in Matzinger (1994, 2001). She points out that there must be discrimination happening that goes beyond the self–nonself distinction described above. For instance:

1 There is no immune reaction to foreign bacteria in the gut or to the food we eat although both are foreign entities.

2 Conversely, some auto-reactive processes are useful, for example against self molecules expressed by stressed cells.

3 The definition of self is problematic—realistically, self is confined to the subset actually seen by the lymphocytes during maturation.

4 The human body changes over its lifetime and thus self changes as well. Therefore, the question arises whether defences against nonself learned early in life might be autoreactive later.

Other aspects that seem to be at odds with the traditional viewpoint are autoimmune diseases and certain types of tumors that are fought by the immune system (both attacks against self) and successful transplants (no attack against nonself).

Matzinger concludes that the immune system actually discriminates "some self from some nonself". She asserts that the Danger Theory introduces not just new labels, but a way of escaping the semantic difficulties with self and nonself, and thus provides grounding for the immune response. If we accept the Danger Theory as valid we can take care of "nonself but harmless" and of "self but harmful" invaders into our system. To see how this is possible, we will have to examine the theory in more detail.

The central idea in the Danger Theory is that the immune system does not respond to nonself but to danger. Thus, just like the self–nonself theories, it fundamentally supports the need for discrimination. However, it differs in the answer to what should be responded to. Instead of responding to foreignness, the immune system reacts to danger. This theory is borne out of the observation that there is no need to attack everything that is foreign, something that seems to be supported by the counter-examples above. In this theory, danger is measured by damage to cells indicated by distress signals that are sent out when cells die an unnatural death (cell stress or lytic cell death, as opposed to programmed cell death, or apoptosis).

Figure 13.3 depicts how we might picture an immune response according to the Danger Theory (Aickelin and Cayzer, 2002c). A cell that is in distress sends out an alarm signal, whereupon antigens in the neighborhood are captured by antigen-presenting cells such as macrophages, which then travel to the local lymph node and present the antigens to lymphocytes. Essentially, the danger signal establishes a danger zone around itself. Thus B-cells producing antibodies that match antigens within the danger zone get stimulated and undergo the clonal expansion process. Those that do not match or are too far away do not get stimulated.

Matzinger admits that the exact nature of the danger signal is unclear. It may be a "positive" signal (for example heat shock protein release) or a "negative" signal (for example lack of synaptic contact with a dendritic antigen-presenting cell). This is where the Danger Theory shares some of the problems associated with traditional self–nonself discrimination (i.e. how to discriminate danger from non-danger). However, in this case,

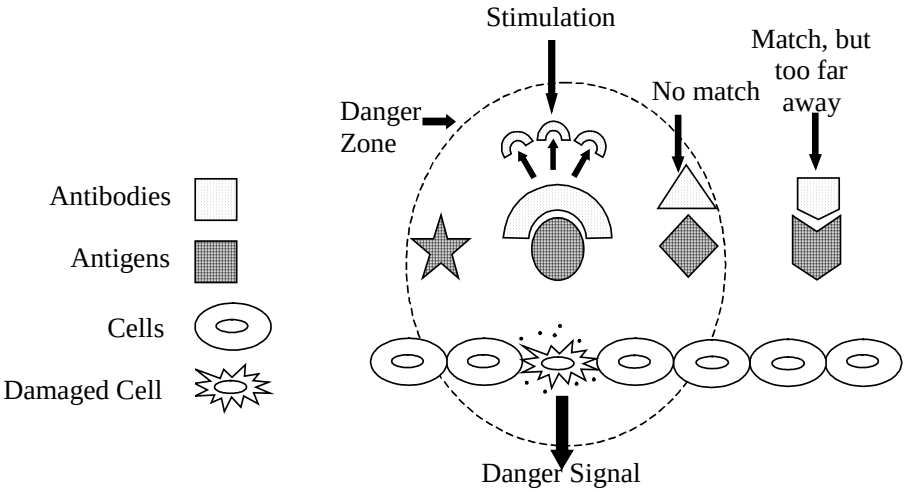

Figure 13.3. Danger theory illustration.

the signal is grounded rather than being some abstract representation of danger.

How could we use the Danger Theory in artificial immune systems? The Danger Theory is not about the way artificial immune systems represent data (Aickelin and Cayzer, 2002c). Instead, it provides ideas about which data the artificial immune systems should represent and deal with. They should focus on dangerous, i.e. interesting, data. It could be argued that the shift from nonself to danger is merely a symbolic label change that achieves nothing. We do not believe this to be the case, since danger is a grounded signal, and nonself is (typically) a set of feature vectors with no further information about whether all or some of these features are required over time. The danger signal helps us to identify which subset of feature vectors is of interest. A suitably defined danger signal thus overcomes many of the limitations of self–nonself selection. It restricts the domain of nonself to a manageable size, removes the need to screen against all self, and deals adaptively with scenarios where self (or nonself) changes over time.

The challenge is clearly to define a suitable danger signal, a choice that might prove as critical as the choice of fitness function for an evolutionary algorithm. In addition, the physical distance in the biological system should be translated into a suitable proxy measure for similarity or causality in artificial immune systems. This process is not likely to be trivial. Nevertheless, if these challenges are met, then future artificial immune system applications might derive considerable benefit, and

new insights, from the Danger Theory: in particular, intrusion detection systems (Aickelin et al., 2003).

13.7 SOME PROMISING AREAS FOR FUTURE APPLICATION

It seems intuitively obvious that artificial immune systems should be most suitable for computer security problems. If the human immune system keeps our body alive and well, why can we not do the same for computers using artificial immune systems? (Aickelin et al., 2004)

We have outlined the traditional approach to do this. However, in order to provide viable intrusion detection systems, artificial immune systems must build a set of detectors that accurately match antigens. In current artificial-immune-system-based intrusion detection systems (Dasgupta and Gonzalez, 2002; Esponda et al., 2004; Hofmeyr and Forrest, 2000), both network connections and detectors are modeled as strings. Detectors are randomly created and then undergo a maturation phase where they are presented with good, i.e. self, connections. If the detectors match any of these they are eliminated, otherwise they become mature. These mature detectors start to monitor new connections during their lifetime. If these mature detectors match anything else, exceeding a certain threshold value, they become activated. This is then reported to a human operator who decides whether there is a true anomaly. If so, the detectors are promoted to memory detectors with an indefinite life span and minimum activation threshold (immunization) (Kim and Bentley, 2002).

An approach such as the above is known as negative selection as only those detectors (antibodies) that do not match live on (Forrest et al., 1994). Earlier versions of negative selection algorithm used a binary representation scheme; however, this scheme shows scaling problems when it is applied to real network traffic (Kim and Bentley, 2001). As the systems to be protected grow larger and larger so does self and nonself. Hence, it becomes more and more problematic to find a set of detectors that provides adequate coverage, whilst being computationally efficient. It is inefficient to map the entire self or nonself universe, particularly as they will be changing over time and only a minority of nonself is harmful, whilst some self might cause damage (e.g. internal attack). This situation is further aggravated by the fact that the labels self and nonself are often ambiguous and even with expert knowledge they are not always applied correctly (Kim and Bentley, 2002).

How can this problem be overcome? One approach might be to borrow ideas from the Danger Theory to provide a way of grounding the

response and hence removing the necessity to map self or nonself. In our system, the correlation of low-level alerts (danger signals) will trigger a reaction (Aickelin et al, 2003). An important and recent research issue for intrusion detection systems is how to find true intrusion alerts from many thousands of false alerts generated (Hofmeyr and Forrest, 2000). Existing intrusion detection systems employ various types of sensors that monitor low-level system events. Those sensors report anomalies of network traffic patterns, unusual terminations of UNIX processes, memory usages, the attempts to access unauthorized files, etc. (Kim and Bentley, 2001). Although these reports are useful signals of real intrusions, they are often mixed with false alerts and their unmanageable volume forces a security officer to ignore most alerts (Hoagland and Staniford, 2002). Moreover, the low level of alerts makes it very hard for a security officer to identify advancing intrusions that usually consist of different stages of attack sequences. For instance, it is well known that computer hackers use a number of preparatory stages before actual hacking. Hence, the correlations between intrusion alerts from different attack stages provide more convincing attack scenarios than detecting an intrusion scenario based on low-level alerts from individual stages. Furthermore, such scenarios allow the intrusion detection system to detect intrusions early before damage becomes serious.

To correlate intrusion detection system alerts for detection of an intrusion scenario, recent studies have employed two different approaches: a probabilistic approach (Valdes and Skinner, 2001) and an expert system approach (Ning et al., 2002). The probabilistic approach represents known intrusion scenarios as Bayesian networks. The nodes of Bayesian networks are intrusion detection system alerts and the posterior likelihood between nodes is updated as new alerts are collected. The updated likelihood can lead to conclusions about a specific intrusion scenario occurring or not. The expert system approach initially builds possible intrusion scenarios by identifying low-level alerts. These alerts consist of prerequisites and consequences, and they are represented as hypergraphs. Known intrusion scenarios are detected by observing the low-level alerts at each stage, but these approaches have the following problems (Cuppens et al., 2002):

1 handling unobserved low-level alerts that comprise an intrusion scenario,

2 handling optional prerequisite actions,

3 handling intrusion scenario variations.

The common trait of these problems is that the intrusion detection system can fail to detect an intrusion when an incomplete set of alerts comprising an intrusion scenario is reported. In handling this problem, the probabilistic approach is more advantageous than the expert system approach because in theory it allows the intrusion detection system to correlate missing or mutated alerts. The current probabilistic approach builds Bayesian networks based on the similarities between selected alert features. However, these similarities alone can fail to identify a causal relationship between prerequisite actions and actual attacks if pairs of prerequisite actions and actual attacks do not appear frequently enough to be reported. Attackers often do not repeat the same actions in order to disguise their attempts. Thus, the current probabilistic approach fails to detect intrusions that do not show strong similarities between alert features but have causal relationships leading to final attacks. This limit means that such intrusion detection systems fail to detect sophisticated intrusion scenarios.

We propose artificial immune systems based on Danger Theory ideas that can handle the above intrusion detection system alert correlation problems (Aickelin et al., 2003). The Danger Theory explains the immune response of the human body by the interaction between antigen-presenting cells and various signals. The immune response of each antigen-presenting cell is determined by the generation of danger signals through cellular stress or cell death. In particular, the balance and correlation between different danger signals depending on different cell death causes would appear to be critical to the immunological outcome. In the human immune system, antigen-presenting cells activate according to the balance of apoptotic and necrotic cells and this activation leads to protective immune responses. Similarly, the sensors in intrusion detection systems report various low-level alerts and the correlation of these alerts will lead to the construction of an intrusion scenario.

13.8 TRICKS OF THE TRADE

Are artificial immune systems suitable for pure optimization? Depending on what is meant by optimization, the answer is probably no, in the same sense as "pure" genetic algorithms are not "function optimizers". One has to keep in mind that although the immune system is about matching and survival, it is really a team effort where multiple solutions are produced all the time that together provide the answer. Hence, in our opinion artificial immune systems are probably more suited as an optimizer where multiple solutions are of benefit, either directly, e.g. because the problem has multiple objectives or indirectly, e.g. when

a neighborhood of solutions is produced that is then used to generate the desired outcome. However, artificial immune systems can be made into more focused optimizers by adding hill-climbing or other functions that exploit local or problem-specific knowledge, similar to the idea of augmenting genetic algorithm to memetic algorithms.

What problems are artificial immune systems most suitable for? As mentioned above, we believe that although using artificial immune systems for pure optimization, e.g. the traveling salesman problem or job shop scheduling, can be made to work, this is probably missing the point. Artificial immune systems are powerful when a population of solution is essential either during the search or as an outcome. Furthermore, the problem has to have some concept of "matching". Finally, because at their heart artificial immune systems are evolutionary algorithms, they are more suitable for problems that change over time and need to be solved again and again, rather than one-off optimizations. Hence, the evidence seems to point to data mining in its wider meaning as the best area for artificial immune systems.

How do I set the parameters? Unfortunately, there is no short answer to this question. As with the majority of other heuristics that require parameters to operate, their setting is individual to the problem solved and universal values are not available. However, it is fair to say that along with other evolutionary algorithms artificial immune systems are robust with respect to parameter values as long as they are chosen from a sensible range.

Why not use a genetic algorithm instead? Because you may miss out on the benefits of the idiotypic network effects.

Why not use a neural network instead? Because you may miss out on the benefits of a population of solutions and the evolutionary selection pressure and mutation.

Are artificial immune systems Learning Classifier Systems under a different name? No, not quite. However, to our knowledge learning classifier systems are probably the most similar of the better known meta-heuristics, as they also combine some features of evolutionary algorithms and neural networks. However, these features are different. Someone who is interested in implementing artificial immune systems or learning classifier systems is likely to be well advised to read about both approaches to see which one is most suited for the problem at hand.

13.9 CONCLUSIONS

The immune system is highly distributed, highly adaptive, self-organizing in nature, maintains a memory of past encounters, and has

the ability to continually learn about new encounters. The artificial immune system is an example of a system developed around the current understanding of the immune system. It illustrates how an artificial immune system can capture the basic elements of the immune system and exhibit some of its chief characteristics.

Artificial immune systems can incorporate many properties of natural immune systems, including diversity, distributed computation, error tolerance, dynamic learning and adaptation and self-monitoring. The human immune system has motivated scientists and engineers for finding powerful information processing algorithms that has solved complex engineering tasks. The artificial immune system is a general framework for a distributed adaptive system and could, in principle, be applied to many domains. The artificial immune system can be applied to classification problems, optimization tasks and other domains. Like many biologically inspired systems it is adaptive, distributed and autonomous. The primary advantages of the artificial immune system are that it only requires positive examples, and the patterns it has learnt can be explicitly examined. In addition, because it is self-organizing, it does not require effort to optimize any system parameters.

To us, the attraction of the immune system is that if an adaptive pool of antibodies can produce "intelligent" behavior, can we harness the power of this computation to tackle the problem of preference matching, recommendation and intrusion detection? Our conjecture is that if the concentrations of those antibodies that provide a better match are allowed to increase over time, we should end up with a subset of good matches. However, we are not interested in optimizing, i.e. in finding the one best match. Instead, we require a set of antibodies that are a close match but which are at the same time distinct from each other for successful recommendation. This is where we propose to harness the idiotypic effects of binding antibodies to similar antibodies to encourage diversity.

SOURCES OF ADDITIONAL INFORMATION

The following websites, books and proceedings should be an excellent starting point for those readers wishing to learn more about artificial immune systems.

1 *Artificial Immune Systems and Their Applications* by D. Dasgupta (ed.), Springer, Berlin, 1999.

2 *Artificial Immune Systems: A New Computational Intelligence Approach* by L. de Castro and J. Timmis, Springer, Berlin, 2002.

3 *Immunocomputing: Principles and Applications* by A. Tarakanov et al., Springer, Berlin, 2003.

4 *Proceedings of the International Conference on Artificial Immune Systems (ICARIS),* Springer, Berlin, 2003.

5 Artificial Immune Systems Forum Webpage: http://www.artificial-immune-systems.org/artist.htm

6 Artificial Immune Systems Bibliography: http://issrl.cs.memphis.edu/ Artificial Immune Systems/Artificial Immune Systems_bibliography.pdf

References

Aickelin, U. and Cayzer, S., 2002c, The danger theory and its application to artificial immune systems, in: *Proc. 1st Int. Conf. on Artificial Immune Systems* (Canterbury, UK), pp. 141–148.

Aickelin, U., Bentley, P., Cayzer, S., Kim, J. and McLeod, J., 2003, Danger theory: The link between artificial immune systems and intrusion detection systems, in: *Proc. 2nd Int. Conf. on Artificial Immune Systems* (Edinburgh), Springer, Berlin, pp. 147–155.

Aickelin, U., Greensmith, J. and Twycross, J., 2004, Immune system approaches to intrusion detection—a review, in: *Proc. ICARIS-04, 3rd Int. Conf. on Artificial Immune Systems* (Catania, Italy), Lecture Notes in Computer Science, Vol. 3239, pp. 316–329, Springer, Berlin.

Amazon, 2003, Recommendations, http://www.amazon.com/

Cayzer, S. and Aickelin, U., 2002a, A recommender system based on the immune network, in: *Proc. CEC2002* (Honolulu, HI), pp. 807–813.

Cayzer, S. and Aickelin, U., 2002b, On the effects of idiotypic interactions for recommendation communities in artificial immune systems, in: *Proc. 1st Int. Conf. on Artificial Immune Systems* (Canterbury, UK), pp. 154–160.

Cuppens, F. et al., 2002, Correlation in an intrusion process, *Internet Security Communication Workshop (SECI'02).*

Dasgupta, D., ed., 1999, *Artificial Immune Systems and Their Applications,* Springer, Berlin.

Dasgupta, D., Gonzalez, F., 2002, An immunity-based technique to characterize intrusions in computer networks, *IEEE Trans. Evol. Comput.* **6**:1081–1088.

de Castro, L. N. and Von Zuben, F. J., 2002, Learning and optimization using the clonal selection principle, *IEEE Trans. Evol. Comput.*, Special issue on artificial immune systems, **6**:239–251.

Esponda, F., Forrest, S. and Helman, P., 2004, A formal framework for positive and negative detection, *IEEE Trans. Syst., Man Cybernet.*, **34**:357–373.

Farmer, J. D., Packard, N. H. and Perelson, A. S., 1986, The immune system, adaptation, and machine learning, *Physica* **22**:187–204.

Forrest, S., Perelson, A. S., Allen, L. and Cherukuri, R., 1994, Self–nonself discrimination in a computer, in: *Proc. IEEE Symposium on Research in Security and Privacy,* (Oakland, CA), pp. 202–212.

Hightower, R. R., Forrest, S. and Perelson, A. S., 1995, The evolution of emergent organization in immune system gene libraries, *Proc. 6th Conference on Genetic Algorithms,* pp. 344–350.

Hoagland, J. and Staniford, S., 2002, *Viewing Intrusion Detection Systems Alerts: Lessons from SnortSnarf,* http://www.silicondefense.com/software/snortsnarf

Hofmeyr, S. and Forrest, S., 2000, Architecture for an artificial immune systems, *Evol. Comput.* **7**:1289–1296.

Jerne, N. K., 1973, Towards a network theory of the immune system, *Ann. Immunol.* **125**:373–389.

Kim, J. and Bentley, P., 1999, The artificial immune model for network intrusion detection, *Proc. 7th Eur. Congress on Intelligent Techniques and Soft Computing (EUFIT'99).*

Kim, J. and Bentley, P., 2001, Evaluating negative selection in an artificial immune systems for network intrusion detection, *Proc. Genetic and Evolutionary Computation Conference 2001,* pp. 1330–1337.

Kim, J. and Bentley, P., 2002, Towards an artificial immune systems for network intrusion detection: an investigation of dynamic clonal selection, *Proc. Congress on Evolutionary Computation 2002,* pp. 1015–1020.

Kubi, J., 2002, *Kubi Immunology,* 5th edn, Richard A. Goldsby, Thomas J. Kindt and Barbara A. Osborne, eds, Freeman, San Francisco.

Matzinger, P., 2003, http://cmmg.biosci.wayne.edu/asg/polly.html

Matzinger, P., 2001, The danger model in its historical context, *Scand. J. Immunol.* **54**:4–9.

Matzinger, P., 1994, Tolerance, danger and the extended family, *Ann. Rev. Immunol.* **12**:991–1045.

Morrison, T. and Aickelin, U., 2002, An artificial immune system as a recommender system for web sites, in: *Proc. 1st Int. Conf. on Artificial Immune Systems (ICARIS-2002)* (Canterbury, UK), pp. 161–169.

Ning, P., Cui, Y. and Reeves, S., 2002, Constructing attack scenarios through correlation of intrusion alerts, in: *Proc. 9th ACM Conf. on Computer and Communications Security,* pp. 245–254.

Perelson, A. S. and Weisbuch, G., 1997, Immunology for physicists, *Rev. Mod. Phys.* **69**:1219–1267.

Resnick, P. and Varian, H. R., 1997, Recommender systems, *Commun. ACM* **40**:56–58.

Valdes, A. and Skinner, K., 2001, Probabilistic alert correlation, *Proc. RAID'2001,* pp. 54–68.